Quick & Dirty Time Series Analysis with Clojure & Incanter

Hi!

I'm Bruce Durling

@otfrom

CTO of green big data startup Mastodon C

www.mastodonc.com

Co-Organizer & Co-Founder
of the *London Clojurians*

Step 0

Log Files
Databases
APIs

People

Step 1

ETL

(this will take longer than you think)

How big is your data?

No, I mean
OMFG BIG

Use Hadoop

clojure-hadoop, cascalog, Hive

GBs or lower?

JFDI

incanter.io
read-dataset

Grouping
Counting
Formatting

incanter.core

dataset
col-names
conj-cols

$ or sel
$group-by
$where

$map
$rollup
$join

clj-time.core
clj-time.format
clj-time.coerce

DATA

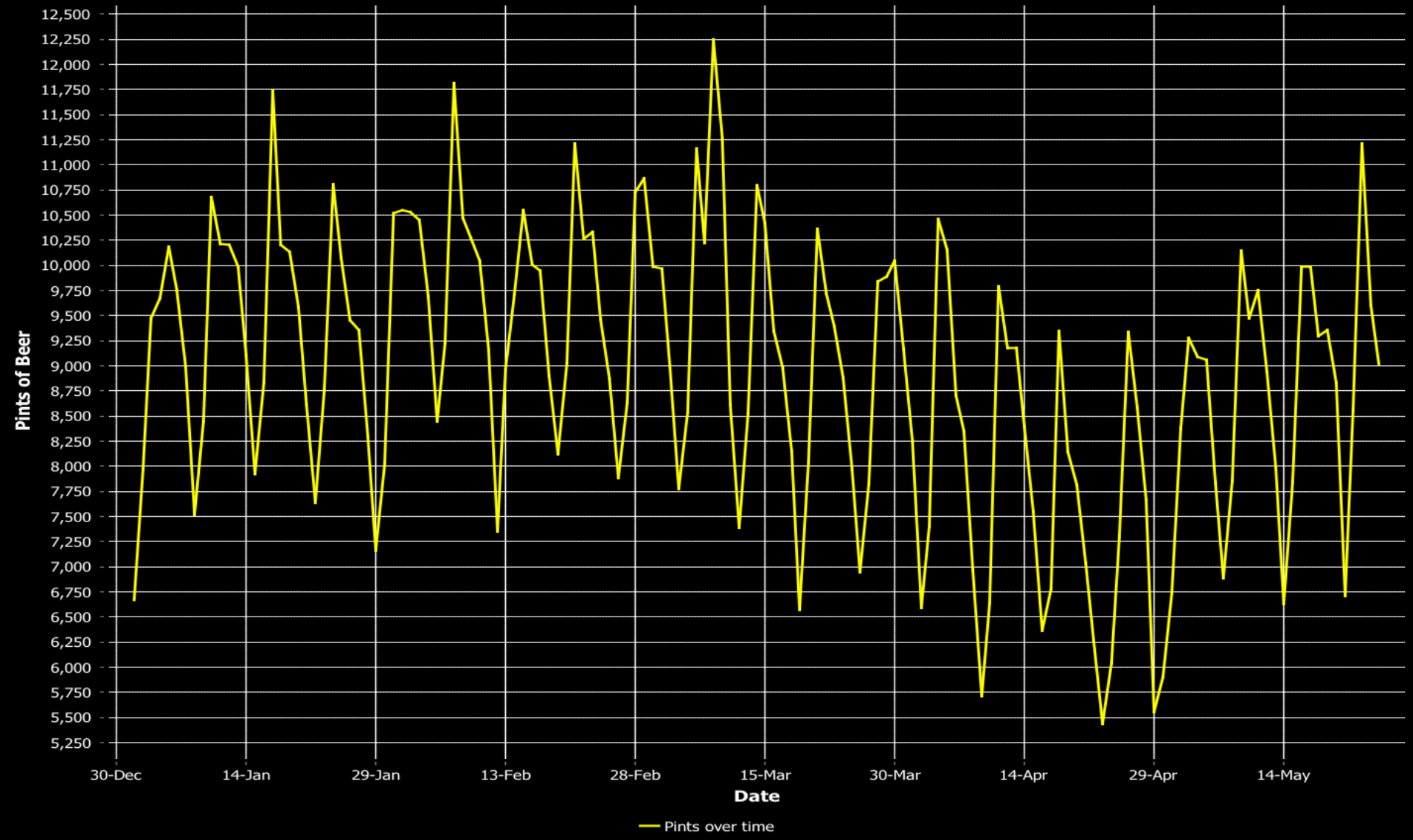
data

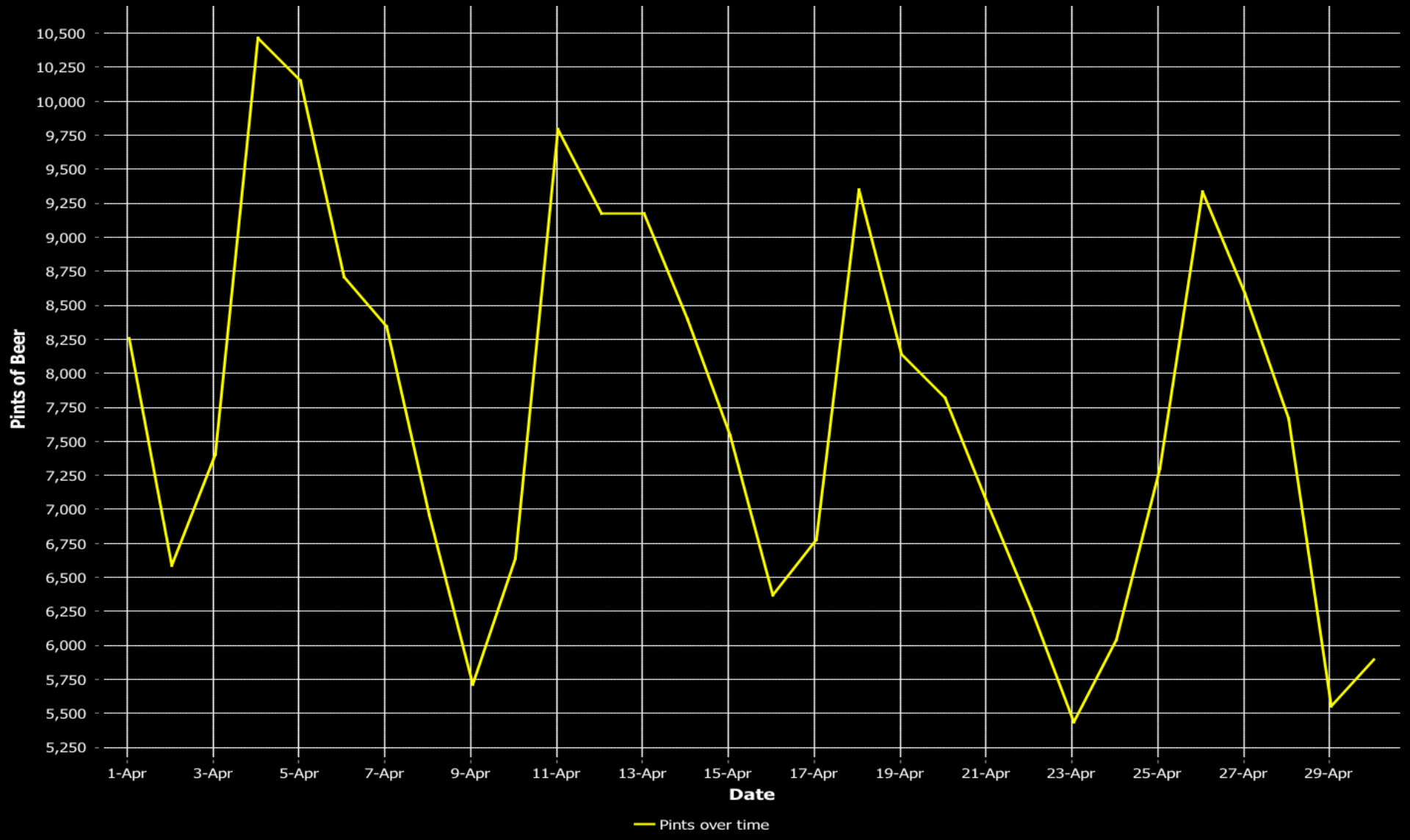
Step 2

Graph It

incanter.charts
time-series-plot

Look for
Extreme Outliers
Noise
Trend
Seasonality


12,500
12,250
12,000
11,750
11,500
11,250
11,000
10,750
10,500
10,250
10,000
9,750
9,500
9,250
9,000
8,750
8,500
8,250
8,000
7,750
7,500
7,250
7,000
6,750
6,500
6,250
6,000
5,750
5,500
5,250
Pints of Beer
30-Dec
14-Jan
29-Jan
13-Feb
28-Feb
15-Mar
30-Mar
14-Apr
29-Apr
14-May
Date
Pints over time


Pints of Beer
10,500
10,250
10,000
9,750
9,500
9,250
9,000
8,750
8,500
8,250
8,000
7,750
7,500
7,250
7,000
6,750
6,500
6,250
6,000
5,750
5,500
5,250
1-Apr
3-Apr
5-Apr
7-Apr
9-Apr
11-Apr
13-Apr
15-Apr
17-Apr
19-Apr
21-Apr
23-Apr
25-Apr
27-Apr
29-Apr
Date
Pints over time

Step 3

Deal with outliers
mean
median
kill

incanter.stats
mean
median

Deal with noise

High Frequency Data can be misleading

incanter.zoo
roll-mean
roll-median
roll-apply

Detrend the data

Difference

```
(defn difference [lag coll]
    (incanter.core/minus
        (drop lag coll) (drop-last lag coll)))
```

Deseasonalise the data

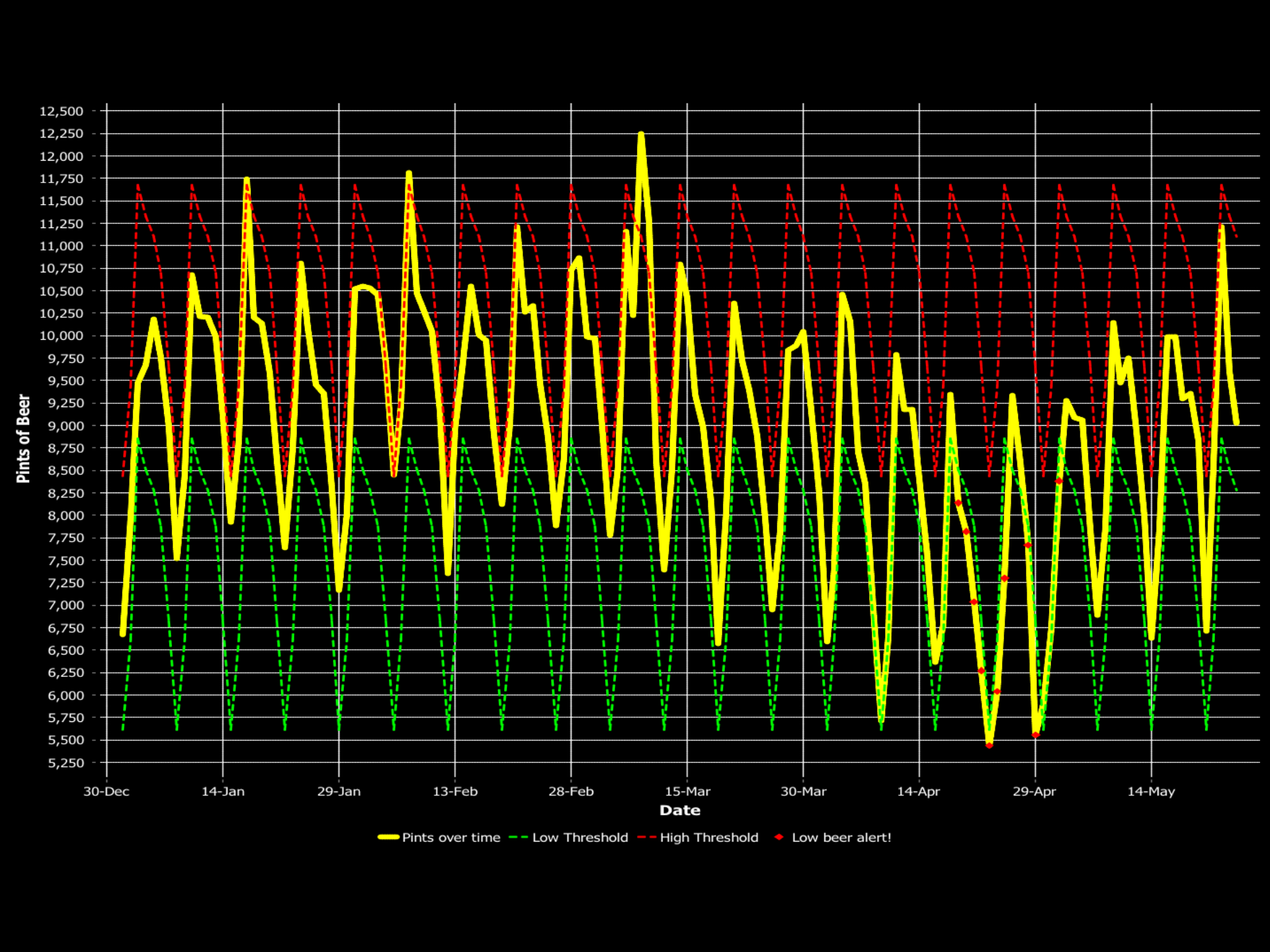
incanter.core
$rollup

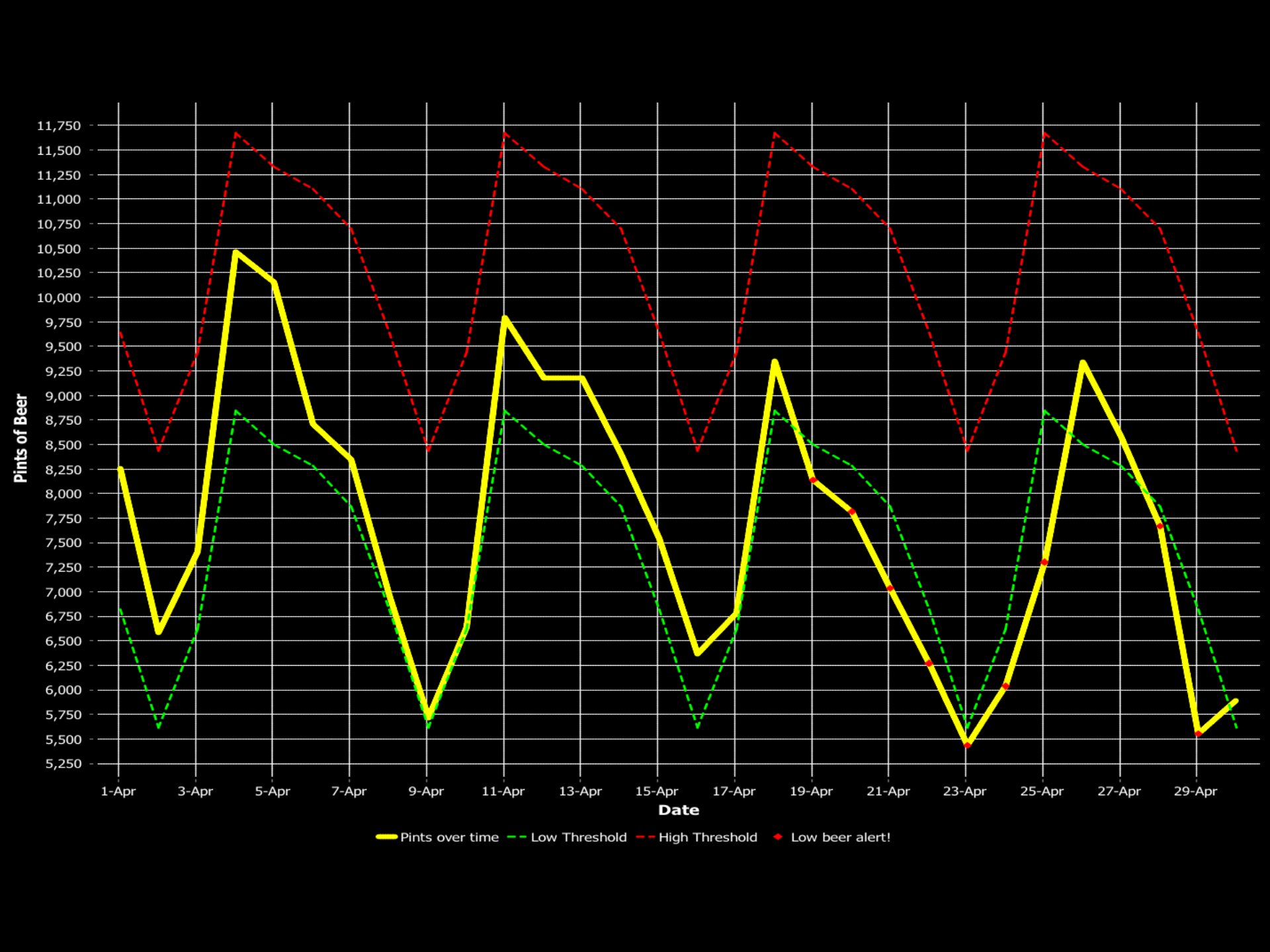
Step 4

Find The Outliers

Standard Deviation

incanter.stats
sd


12,500
12,250
12,000
11,750
11,500
11,250
11,000
10,750
10,500
10,250
10,000
9,750
9,500
9,250
9,000
8,750
8,500
8,250
8,000
7,750
7,500
7,250
7,000
6,750
6,500
6,250
6,000
5,750
5,500
5,250
Pints of Beer
30-Dec
14-Jan
29-Jan
13-Feb
28-Feb
15-Mar
30-Mar
14-Apr
29-Apr
14-May
Date
Pints over time
Low Threshold
High Threshold
Low beer alert!


Pints of Beer
11,750
11,500
11,250
11,000
10,750
10,500
10,250
10,000
9,750
9,500
9,250
9,000
8,750
8,500
8,250
8,000
7,750
7,500
7,250
7,000
6,750
6,500
6,250
6,000
5,750
5,500
5,250
1-Apr
3-Apr
5-Apr
7-Apr
9-Apr
11-Apr
13-Apr
15-Apr
17-Apr
19-Apr
21-Apr
23-Apr
25-Apr
27-Apr
29-Apr
Date
Pints over time
Low Threshold
High Threshold
Low beer alert!

Step 5

Real Time Alerts

```
(rand-nth
 [:riemann
  :nimrod
  :storm])
```

Step 6

Iterate

Formulate a Hypothesis

Get the data you need

(step-0 previous-results new-data)

Thank You

Mastodon C

"Herds of intelligent roaming Hadoop jobs, migrating from datacentre to datacentre, in search of cold iron."
@mrchrisadams

@MastodonC

www.mastodonc.com